Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1-16 (Cancelled).

17. (Currently Amended) A method for utilizing shared resources in a computerized system at a command level, with the aid of a processor for processing a plurality of commands and executing thereof using at least two ~~or more~~ of said shared resources, wherein at least one command of the plurality of commands comprises at least two ~~or more~~ sub-commands to be executed at different said at least two ~~or more~~ shared resources for executing said at least one command, the method comprising steps of:

[[-]] deriving, from each of said plurality of commands, subcommands respectively related to said shared resources,

[[-]] assigning priorities to said subcommands,

[[-]] forwarding said subcommands to one or more queues of the respective two or more shared resources, so that each of said queues comprises the subcommands related to a particular shared resource, executing the subcommands from said queues by said shared resources in an asynchronous manner, and according to said subcommand priorities by each of the shared resources, by allowing at least one ~~subcommands~~ subcommand of one command to ~~start executing~~ start executing while subcommands of another command are not finished executing,

wherein the step of assigning priorities to the subcommands comprises assigning a combined priority to each subcommand, the combined priority being determined based on the priority of the subcommand in the command and the priority of said command, so that commands having a higher priority, have higher priority sub-commands.

18. (Previously presented) The method according to Claim 17, further comprising a step of assigning different command priorities to said commands, wherein the command priorities set an order of their urgency.

19. (Previously presented) The method according to Claim 18, wherein the step of assigning priorities to said subcommands comprises assigning to them the priority equal to that of the command from which the subcommands are derived.

20. (Previously presented) The method according to Claim 17, wherein the step of assigning priorities to said subcommands comprises defining one group of the subcommands as

critical subcommands for execution of their respective commands, and another group of the subcommands as non-critical commands for execution of their respective commands, wherein priorities of the critical subcommands are higher than priorities of the non-critical subcommands.

21. (Cancelled).

22. (Currently Amended) The method according to Claim 17, further comprising steps of:

[[-]] in each of the shared resources, upon executing the subcommands from the subcommand queue according to the subcommand priorities, obtaining respective responses of successful completion and outputting thereof into a response queue of the shared resource;

[[-]] forwarding each of the responses from the response queues to the command from which the corresponding subcommand was derived, for further creating reports of successful completion relating to said commands.

23. (Currently Amended) The method according to Claim 22, further comprising a step of issuing a preliminary report with respect to a particular command before completing its execution, but upon receiving, with respect to said particular command, of at least one ~~or more~~ of said responses of successful completion concerning the respective subcommands

having high priority, in order to initiate urgent execution of another command of said plurality.

24. (Currently Amended) A computer control system for utilizing shared resources at a command level, the computer control system comprising at least one ~~or more~~ command ~~processors~~ processor for processing a plurality of commands, each of said command processors being capable of cooperating with at least two ~~or more~~ said shared resources; each of said command processors being operative to:

[[-]] derive, from a command of said plurality of commands, at least two ~~or more subcommands to be~~ respectively executed subcommands at said at least two ~~or more~~ shared resources,

[[-]] assign priorities to said subcommands,

[[-]] forward the at least two ~~or more~~ subcommands of said command to the respective at least two ~~or more~~ shared resources for execution, wherein subcommands of other commands ~~being~~ are also forwarded to said at least two ~~or more~~ shared resources for execution;

[[-]] receive from said shared resources responses of successful completion concerning the respective subcommands, and

[[-]] based on the responses concerning said subcommands, form reports of successful completion or partial reports concerning the respective commands,

[[-]] thereby enabling said at least two ~~or more~~ shared resources to execute the subcommands of different commands in an asynchronous manner, according to the priorities of said subcommands, by allowing subcommands of one command to start execution while subcommands of another command are not finished executing,

wherein the step of assigning priorities to the subcommands comprises assigning a combined priority to each subcommand, the combined priority being determined based on the priority of the subcommand in the command and the priority of said command, so that commands having a higher priority, have higher priority sub-commands.

25. (Currently Amended) The computer control system according to Claim 24, further comprising a ~~master~~ processor capable of cooperating with said command processors, wherein said processor has a higher level with respect to said command processors~~ being slave processors~~;

said master processor being operative to distribute the commands between said command processors, and receive from said command processors reports of successful completion concerning the respective commands.

26. (Currently Amended) The computer control system according to Claim 25, wherein the ~~master~~ processor is operative to sort the commands by priorities between said command processors.

27. (Currently Amended) The computer control system according to Claim 24, wherein each of said command processors is capable of dividing said subcommands into a group of critical subcommands being critical for execution of their respective commands, and a group of non-critical subcommands being non-critical for execution of their respective commands, wherein priorities of the critical subcommands are higher than priorities of the non-critical subcommands.

28. (Currently Amended) The computer control system according to Claim 27, wherein at least one of said command processors is capable of issuing a preliminary report with respect to a particular command of said plurality, before the particular command is completely executed, the preliminary report is based on at least one ~~or more~~ of said responses of successful completion concerning the critical subcommands of the particular command.

29. (Currently Amended) The computer control system according to Claim~~s~~ 24, ~~additionally~~ further comprising:

at least two ~~or more~~ input memory buffers respectively associated with said at least two ~~or more~~ of the shared resources, for gathering and queuing said subcommands of different commands to be input to the shared resource, and

at least two ~~or more~~ output memory buffers for queuing responses when outputted from the respective shared resources.

30. (Currently Amended) The computer control system according to Claim 29, wherein said input memory buffers are capable of sorting the subcommands in ~~the~~a queue so that the first subcommand to be read from the queue is always that having the highest priority in the queue.

31. (Currently Amended) The computer control system according to Claim 24, ~~being~~ wherein the control system is a system for controlling a telecommunication network.

32. (Currently Amended) A computerized system with shared resources, comprising the computer control system according to Claim 24.

33. (Currently Amended) A method for utilizing shared resources at a command level in a computerized system comprising a processor for processing commands, and at least

one ~~or more~~ shared ~~resources~~ resource required for execution of said commands, the method comprises steps of:

[[-]] deriving, from each of said commands, subcommands respectively related to said at least one ~~or more~~ shared ~~resources~~resource,

[[-]] assigning priorities to said subcommands, forwarding said subcommands to at least one ~~or more~~ input ~~queues~~ queue of the ~~respective~~ at least one ~~or more~~ shared resources, respectively, so that each of said input queues comprises the subcommands related to a particular shared resource and having their assigned priorities,

[[-]] executing the subcommands from each of said queues according to said subcommand priorities by each of the shared resources in an asynchronous manner, by allowing at least one ~~or more subcommands~~ subcommand of one command to start executing while subcommands of another command are not finished executing,

wherein the step of assigning priorities to the subcommands comprises assigning a combined priority to each subcommand, the combined priority being determined based on the priority of the subcommand in the command and the priority of said command, so that commands having a higher priority, have higher priority sub-commands.

34. (Currently Amended) A computer control system capable of implementing ~~the~~ a method ~~according to Claim 33~~for utilizing shared resources at a command level in a computerized system comprising a processor for processing commands, and at least one shared resource required for execution of said commands, the method comprising steps of:

deriving, from each of said commands, subcommands respectively related to said at least one shared resource,

assigning priorities to said subcommands,

forwarding said subcommands to at least one input queue of the at least one shared resources, respectively, so that each of said input queues comprises the subcommands related to a particular shared resource and having their assigned priorities,

executing the subcommands from each of said queues according to said subcommand priorities by each of the shared resources in an asynchronous manner, by allowing at least one subcommand of one command to start executing while subcommands of another command are not finished executing

wherein the step of assigning priorities to the subcommands comprises assigning a combined priority to each subcommand, the combined priority being determined based on the priority of the subcommand in the command and the priority of said command, so that commands having a higher priority, have higher priority sub-commands.